EDP ANALYZER

MAY, 1970
VOL. 8, NO. 5

DATA SECURITY IN THE CDB

Concluding our four-report series on the corporate data base (CDB), we consider the question of data security. When highly sensitive data is in on-line files, real security requires on-line users and the computer to be in a guarded, shielded room. At the other extreme, how much security does a system provide that serves many remote terminals over telephone lines, using a regular operating system and without protected communications? The answer: against a skilled penetrator with needed resources, not very much. There are useful solutions short of the guarded, shielded room, particularly when the data is not "top secret." Practical data security systems can be created – but sad to say, not easily. Here is a look at where the field stands today.

Continental Airlines installed in mid-1968 a modern, computer-based reservation system that they call SONIC 360. SONIC 360 is based on the IBM PARS – Programmed Airlines Reservation System. The central computer is located in Los Angeles, and it serves Continental offices ranging from Chicago to Honolulu.

Currently, SONIC 360 handles about 400,000 passenger itinerary records, plus the seat inventory for Continental's 200-plus daily flights for eleven months into the future, plus seat availability on some 1800 flights of other connecting airlines. There are 500 agent sets located in 26 cities, using typewriter-like terminals. In addition, CRT terminals are used at Continental's five major central reservation offices.

Since the file of passenger records and seat inventory is so crucial to Continental's business, data security is an important issue. When an employee who will use SONIC 360 logs in for the day, he gives his personal identification (name and number) and his duty code; he is urged not to disclose his identification numbers to others. The terminal identification is picked up automatically. The duty code is a two character code for the type of duty—RC (central reservations), AS (sales agent), SU (supervisor), TA (training agent), etc. The entry is checked against security tables in the system which list names, numbers, duty code, type(s) of terminals authorized to use, and types of transactions authorized to enter.

For each transaction that an employee enters during his working day – query about space availability, request for seat, new passenger record, etc. – he both signs in and signs out using his identification number. He signs out so that he will not be responsible for any later transactions. All transactions are first checked for validity and authority, then automatically processed and logged. Such audit

trail logs can provide an after-the-fact record of security violations that the system did not detect. To date, there have been no indications of such violations.

To perform the authority check, the system refers to the security tables that indicate which employee-and-terminal combinations are authorized to: (a) enter passenger records, (b) change a flight schedule, (c) modify a security table, etc. If a violation occurs and has been programmed as potentially harmful, supervision is notified.

The typewriter-like agent sets are located at Continental ticket counters, staffed by Continental personnel. They are locked with a key when the office is closed. The CRT terminals are all located at the company's central reservation rooms.

In the first 18 months of operation, only two security violations occurred. In one case, an unauthorized broadcast message was entered into the system from an agent set. It was detected as unauthorized by the system and supervision was signalled. It took only 15 minutes for Operations to retrieve the message from the log, determine the city and the terminal, and call the supervisor at that office. The terminal had been unattended for a short time, but the supervisor was able to determine which unauthorized person had had an opportunity to operate it.

Data security for airlines reservations systems is a necessary function. Unauthorized disclosure of passenger records and/or seat inventory records could be troublesome. Unauthorized *changing* of these records would be quite serious. Continental's management feels that their data security system is providing about the right degree of protection for their needs.

Penetration of systems

To assess the protection afforded by today's systems against a skilled penetrator, consider the experiences of Professor E. L. Glaser, of Case Western Reserve University in Cleveland, Ohio. Professor Glaser has had a wealth of experience in the computer field, including years at Project MAC at M.I.T. Data security is one of his pet subjects. To put it bluntly, Professor Glaser is a skilled penetrator. People developing data security systems use his services to try to penetrate their systems, as a test of their security measures. After learning the standard operating procedures for the system, and thinking about the matter for a time, it is not unusual for him to be able to break through the security measures within five minutes time at a terminal.

The trouble is, as Professor Glaser points out, that the system designers may leave one or more "trap doors" through which a penetrator can enter. A trap door is some idiosyncrasy of the software (or hardware-software combined) that provides an opportunity to penetrate or bypass the security controls. Sometimes trap doors are inserted intentionally as an aid to debugging — and someone forgets to close them. Or they might be built intentionally to permit future penetration. Or they may occur accidentally; for one reason or another, the designers were not aware that they existed.

In one system that Professor Glaser tested, the security measures for on-line operations were quite good. But a user could also initiate batch-type jobs, to operate in the background mode, from his remote terminal. So Professor Glaser wrote a simple job — 12 lines of code — to operate in the background mode. This job went through every file in the system, found the owners' names and passwords, extracted all passwords, and stored them in a new file. Further, the program then scrambled the passwords so that no one stumbling on the file would recognize the contents. The program then unscrambled the password and printed them out on his terminal. Total time: one evening of his time thinking about the program, about one minute at the terminal to enter the program, and five minutes of computer time.

In another case, Professor Glaser was asked to test the security measures of a new commercial time sharing system. He found that the security measures were excellent — but the operating system in which they were embedded had all sorts of trap doors. Within five minutes at the terminal, he was able to go around the security measures and cause the whole system to "crash" — come to an abrupt, disorderly halt.

We will have more to say in this report about Professor Glaser's views on data security. Suffice it to say, he is skeptical about the ability of any of today's remote terminal systems to withstand the efforts of a skilled penetrator, if the data in the files is valuable enough to warrant the effort.

The ADEPT-50 *system*

The ADEPT-50 time sharing system, developed by System Development Corporation, in Santa Monica, California, was designed with data security requirements very much in mind; see Reference 1. It was developed by SDC for the Department of Defense, to operate on a slightly modified IBM 360/50. (The modification was to add the Read Protect feature.) It is being evaluated at a number of military installations around the country.

ADEPT-50 was designed to serve users at remote terminals that can communicate with the computer over the telephone dial network. Further, these users can write their own programs in a variety of programming languages, including assembly language. For handling defense classified information, DoD policy recommends that both users and the computer be in a guarded, shielded room. In the case of ADEPT-50, however, different installations may choose equivalent controls with fewer physical constraints, depending on the classification of the information. In some instances, shielded lines to remote terminals may be used, or encrypting methods may be used, to ease the above constraints.

We will have more to say about the ADEPT-50 security measures later in this report. The point to be made here is: even with the most advanced data security measures available in ADEPT-50, still its use is restricted to guarded, shielded rooms when the data is sufficiently sensitive, due to the uncertainty about today's computer-based security measures. Companies that are thinking of putting very private data into on-line files, where the computer serves remote terminals, should keep that fact in mind.

The problem of data security

In this issue, we will attempt to give an overview of the data security problem. The subject is sufficiently complex that we will have to refer to sources of detailed information at numerous points in the discussion.

First of all, just what is data security? The definition that we like best, and one that we have seen in several sources, is: the protection of data from accidental or intentional but unauthorized modification, destruction, or disclosure. Note that we are not addressing the important subject of privacy – which involves having "too much" data about a person or a company in a data file, and where even authorized access might endanger the person's right to privacy.

Further, we are considering primarily those computer-based systems that use remote terminals. Security problems can arise in batch-type systems that use no remote terminals. But the really challenging problems are arising with remote terminal systems employing multiprogramming. Note that these systems are not limited to ones using telephone lines for communications; private communication lines are also included.

The problem of data security will be nontrivial with almost any remote terminal system. It will become much more important as a corporate data base (CDB) is installed – because of the tendency to put more sensitive data in the CDB.

Types of threats

A major contribution to the subject of data security was a session held at the 1967 Spring Joint Computer Conference. The Proceedings of that conference (Reference 2) are must reading for any student of the subject.

Dr. Willis Ware, of the Rand Corporation, Santa Monica, California, organized the session and discussed some types of threats. H. E. Petersen and R. Turn, also of the Rand Corporation, gave a list of threats. We will summarize briefly from these papers:

TYPES OF THREATS

Accidental

1. A user error, where he stumbles upon a "trap door."
2. A system error, either hardware or software, caus-

ing a failure of one or more of the protective features.

3. A communication "error" from cross talk or a bad switching mechanism.
4. Accidental revealing of protective features to computer operators during recovery period, or to maintenance engineers during maintenance.

Deliberate, passive

5. Electromagnetic pickup from terminal, communications line, computer, or peripheral equipment.
6. Wire tapping of the communications lines.
7. Unauthorized person looking at terminal printout.

Deliberate, active

8. "Browsing" through a file, looking for sensitive data.
9. Impersonation of an authorized user.
10. "Between lines" entry – tying into the system while a user is signed on but temporarily inactive.
11. "Piggy backing" – intercepting messages between authorized user and computer; substituting messages; cancelling sign off.
12. Entry into the system by people who understand the safeguards and how to get around them – system programmers, operators, maintenance personnel, managers, former employees; planting of entry points.
13. Entry via a "trap door."
14. Reading of residual data – from memory, tapes, disks, printer ribbons, printer platens.

It is not always realized just how much electromagnetic radiation computing equipment gives off. We were recently told of a test where a van parked next to an unshielded computer center. In the van was equipment for receiving and processing the radiated signals. A high speed printer, driven by these signals, produced the same output as was printed in the center.

A report on message interception was prepared for the California Senate in 1957, to indicate the widespread use of eavesdropping techniques (Reference 3). It was reported that in 1955, a good number of labor leaders requested the telephone companies to check for wire taps on their lines, and were willing to pay for the inspections. Of some 200 checks conducted, taps were found in about 70% of the cases. In a sampling of about 100 other people, about 25% were found to have taps. The report also tells of public agency usage of wire taps. Court orders to install taps were most common – not for extreme crimes such as homicide or narcotics – but for gambling and prostitution, historically the crimes where protection has most frequently been available for a price. The point is, of course, if enough money is involved, wire taps will occur.

Passive penetration may be quite costly compared to the value of the information received; active penetration can be much more damaging. Either or both can be used readily with today's remote terminal systems.

The risk for business data

Typically, the types of data files that have been first converted to the computer have been the following:

1. Product data files, including inventory data, manufacturing data, and such.
2. Customer and supplier files, which hold open order data, history of past purchases, etc.
3. Financial files, including accounts payable, accounts receivable, and other general ledger accounts.
4. Personnel files, which hold both personal and payroll data.

As far as fear of disclosure is concerned, inventory files are perhaps the least sensitive. While the company would not want the information disclosed, it would probably not be hurt too badly if the information were disclosed. Of course, if the information were maliciously changed, that would be a different matter. Fast response order entry systems, such as airline reservations systems, are of this type.

More sensitive data is typified by personnel files or customer files. Rate-of-pay data has traditionally been protected from unauthorized access. And a company surely would not like for its customer open order data and history of purchases data to fall into the hands of competitors.

The most highly sensitive data is represented by (a) plans for new products, entering new markets, submitting of competitive bids; (b) the financial condition of the company, including detailed lists of assets, liabilities, taxes, etc.; (c) trade secrets, patent applications, letters of complaints about company products or serv-

ices; and (d) data about the security system itself, such as lists of passwords.

Mr. R. H. Courtney, of IBM, at a panel session of the 1969 Fall Joint Computer Conference, pointed out the tradeoffs facing designers of security systems. The data has a value to its owner, and a cost of providing security. These factors must be balanced against the value of the data to an intruder, and the cost of gaining access. The greater the value of the data, the more an intruder will pay to gain access. Value of the data is a function of both its quality and quantity.

Also, as Courtney pointed out, the intruder may not need to access the data itself. He may achieve his purpose if he denies access to the rightful users of the data — by locking up files, or by "crashing" the system, for instance.

P. L. Schiedermayer, of the Security Engineering Company of California, at this same 1969 FJCC panel, discussed the shortcomings of most business security systems. "Business is inclined to put all of its most sensitive data into one system at one time," he says — "something the military would never do." Further, they do not make use of many proven safeguards — security checks of personnel, locked computer rooms, and so on. And the incipient threat to business data may in fact be greater than to military data; there are more people in a position to spy on it.

And Professor Glaser adds a further observation. When passwords are used, two mistakes commonly occur. First, "obvious" passwords are used — dates of birth, street address numbers, etc. Second, passwords are stored in obvious places — like the homeowner who "hides" the key under the doormat or in the mail box.

Types of users

There are normally many types of personnel associated with a remote terminal system. There are the operating personnel, such as airline sales agents, and managers and staff members who are cleared to use the system. There are system programmers, computer operators, computer maintenance people, application programmers, and installation managers in the data processing department. And there are auditors and security personnel who are authorized to probe the system.

As Clark Weissman of SDC points out, the more capability a user has to write programs (and thus use the power of the computer to subvert the system), the more chance he has to violate security provisions. When programming, the closer he gets to working with machine binary code, the greater is the risk.

There are other types of constraints that can be imposed upon users. Some users may be allowed only to retrieve data, others only to store data, while others can both retrieve and update. Some will have the authority to not only use data files themselves but also to authorize others to use the files. Others will have the ability to override security restrictions.

The extreme cases

Dr. Ware and Professor Glaser sketched for us the characteristics of the "how-*not*-to-do-it" case under the present state of the art. First, the software — and particularly the operating system — is so complex that no one person can comprehend it; it is "dirty." The system has a variety of types of users, with different security privileges, read/write privileges, and programming privileges. The system has many remote terminals, of different types, and access to a terminal is easy. Finally, the on-line files hold highly sensitive data. In such an environment, they say, any attempt at data security is doomed from the outset. A skilled penetrator can break in with little difficulty.

A little reflection will show that, unfortunately, the bulk of today's remote terminal systems come very close to this situation.

How about the other extreme, where security is possible? Here are the characteristics. The software must be "clean"— well designed, so that a person can comprehend and verify it for logical consistency and completeness. The security system should be designed as part of the system, not added as an afterthought. There should be a relatively small number of authorized on-line users, each of whom has been given a security clearance. Access to all terminals should be strictly controlled. All

components should be checked for electromagnetic radiation and shielding employed where necessary. If sensitive data is to be transmitted over telephone lines, then some form of communications protection should be used.

Types of countermeasures

The upshot of this discussion is that a data security problem *does* exist in the business environment. If a company is considering putting sensitive data into on-line files in a remote terminal system, then it must be concerned about data security. We will briefly discuss the following types of countermeasures that can be considered for the security system:

- Access management
- File design
- Hardware/software techniques
- Communications protection
- Reliability, auditability, integrity
- General security procedures

Some of the countermeasures are quite costly and would be considered only if the value of the data is very high. It is the user's decision, of course, on just how much security he needs to protect against the threats he considers possible.

The main references to this subject in the technical literature are the 1969 Fall Joint Computer Conference Proceedings (Reference 1), the 1967 SJCC Proceedings (Reference 2), a paper by L. J. Hoffman (Reference 4), and a pamphlet issued by IBM (Reference 5). We will draw heavily on this literature, plus occasional specific references to other sources.

Access management

There are several possible levels of access control. The simplest and most basic is to ask the user to identify himself when he requests service on a remote terminal system. The user must identify himself in a manner acceptable to the system.

The next higher level – again basic and widely used – is to make the user *authenticate* this identification. A common means for this step is for the user to supply a password; the computer checks to see if the password is valid.

This password technique is fairly powerful, but may not really provide a lot of security, particularly passwords that are used over and over. Passwords can be compromised by stealing, by overhearing, by wiretap. Or perhaps the passwords are poorly chosen, so that the intruder can uncover them with a few attempts. Or the intruder may use a computer as his terminal and have it try one combination after another until it finds the right one – unless the security system protects against this.

The password system can be modified to add to its strength. Passwords can be assigned by the system, with characters assigned randomly. Passwords can be changed *aperiodically*, so that the intruder does not know how long he can use a password, or how long he has to uncover it. Passwords may be used one time only, so that each user (and the computer) has a list of his own passwords – with each user having a different list.

Another modification is for the computer to transmit a password and have the user perform some simple but non-obvious transformation on the password and transmit it back to the computer. If a different password were used by the computer each time, it would be difficult for an intruder to learn much about the password system via wiretap.

A fairly common procedure is for passwords to be entered via a non-printing mode of operation of the terminal. This procedure reduces the chance of inadvertent observation. The password would still be revealed to a wiretapper, though.

The next higher level of access control is to require terminal identification, and to restrict users to one (or at most, a few) terminals. Terminal identification can be impersonated – but it makes intrusion harder.

The next higher level of access control is to store security tables in the computer which define the privileges of each user. For instance, the tables might list the files to which the user has access.

Hsiao (Reference 6) developed a pilot model of a time sharing system capable of han-

dling private and public files, for his doctoral thesis. It was implemented at the University of Pennsylvania. Hsiao used an authority item for each user, which listed all files to which the user had access. This authority item showed which files the user owned, which ones he shared ownership of, and which ones he could only use. The user would be denied access to any file not listed in his authority item. Also, for each one of these files to which he had access, additional information was listed. Any portions of the files to which access was blocked for this user were indicated. Also, the mode of use was listed for each file — read only, write only, read and write. Hoffman (Reference 4), in commenting on Hsiao's system, points to the potential large number of entries in the authority items and reflects that this "overhead" may be too high in many instances — where "overhead" is the data and processing used only to provide security.

Weissman (Reference 1) provides the most comprehensive access control system that we came across in our study. His paper describes a formal set theoretic model of a security system, and then goes on to tell how the model was implemented in the ADEPT-50 system. He identifies four types of security objects — users, terminals, jobs, and files. And he identifies three types of security properties — authority, franchise, and category. Authority relates to the levels of security classification. Franchise means "need to know." Category permits need to know designation by special group identification — executive payroll only, etc.

In the ADEPT-50 system, the user identifies himself during the log-in procedure, and provides a password to authenticate this identification. Passwords are used only once, so each user has a list of passwords. Further, users are restricted to using specified terminals. Security tables in the system define the users, terminals, jobs, and files, in terms of authority, franchise, and category. What this means is that each security object — user, terminal, job, file — has a *security profile.*

In ADEPT-50, the authority property has four values — unclassified, confidential, secret, and top secret. The category property has sixteen values, assigned by the using agency. In connection with the franchise of files, there are three types of files: public, private, and semi-private. Public files have no need-to-know list; anyone who passes the authority and category constraints has access. Private files also have no need-to-know list; they are for the private use of their owners (creators). Only the semi-public files have need-to-know lists—lists of all users who are authorized access to the files. Note that access is controlled at the file level, not at the record, segment, or field level. The reason is simply that in the ADEPT-50 implementation, it was felt that any lower level control would be too expensive in overhead — particularly since users could discipline themselves to store similarly classified and logically related data in a single file.

Franchise for a file is more extensive than just implied; it also includes quality of access — read only, write only, read and write, and read-and-write with ability to override the lockout used to avoid simultaneous use.

How costly are the security features of ADEPT-50? Weissman estimates that the security portions of ADEPT-50 required about 5% of the total design time (in man-hours) and about 10% of the coding. About 80% of the code in the security portions is local to just five components of the total system. About 2% of the CPU time is spent in performing security checks.

Some additional comments are in order, in connection with access control. For one thing, with tight access control, it falls to the file owners to purge obsolete data. There may be no way for system managers to know which data is not being used. At the same time, there probably is a need to store the list of passwords at some trustworthy place. If users forget or lose their passwords, some means is needed to get them back into operation and to recover their data. This "trustworthy" place then becomes a potential weak point in the system, a place for an intruder to attempt to break in.

The shortcomings of passwords to authenticate the user's identity have been discussed. New techniques such as the automatic recognition of fingerprints and voice prints are being mentioned. While such methods may make impersonation harder, they would not completely solve the problem. If a password were

used over and over, a tape recording of a user's voice may allow entry. Even more basic, such input must be converted to electrical signals for transmission to the computer – and it is these signals that may be impersonated.

A final comment on access control, based on Babcock's paper in Reference 2. Babcock reported that, with their time shared system, they learned to abort a user and disconnect him as soon as his messages caused validity or authority checks to fail. If such a user is not cut off immediately, then his output tends to go to other users, or vice versa.

File design

Data security can affect file design in several ways. We will discuss the level of access control, the physical separation of files, and some reliability factors.

Level of access control

Access control can be imposed at any of several data levels – from the file level down to the field level.

File level control is the easiest to implement. One or more security tables in the system define the conditions of use for each file. A request for use is checked against the tables; if it passes the tests, access is granted.

One main problem with file level control is that a user – if he is updating the file – can tie up the file and deny access to others for an extended period of time. Even read-only operations are usually not permitted for data being updated. Another problem is that control is really too gross. For instance, a number of users might be authorized access to personnel data, but only a few users would have access to the payroll portion of the data.

Index level control would be the next level. Associated with each entry in the index would be the conditions of use for the related records. Note that the overhead is higher than in the case of file level control.

Record level control would amount to about the same thing as index level control. It might be used with sequential files for which no index was available. Record level control makes sense for *documents* in an information storage and retrieval system.

Segment level control would define access conditions for each data segment – groups of related data fields, such as the salary and earnings segments of a personnel record. This level of control is intuitively appealing. Note, however, that the overhead can become quite high. It would be impractical, for instance, to have a list of authorized users for each segment of each record in the file. It probably would be necessary to set up categories of segments and categories of requests. Segment level control would solve the personnel-payroll file problem mentioned above.

Field level control, for specific fields within a record, has been proposed. The overhead for this type of control might well be horrendous. We suspect that segment level control will prove to be more practical.

As control moves from the file level toward the field level, the overhead – in terms of storing security tables and in processing time – rises rapidly. But one user can tie up – and thus deny access for other users – smaller and smaller portions of the data when that user is updating the data.

Regardless of *level*, other factors enter into access control. These factors include the type of file (public, private, semi-private), mode of use (read only, write only, read and write), and constraints such as clearance. We have discussed these factors earlier in this report.

Physical separation of files

The physical separation of files, and storing them on removable units, makes sense from a security standpoint. With demountable files, not all of them need be on-line simultaneously. Particularly sensitive files might be mounted only at specific times of the day or week, when other controls can be strengthened. Hardware protection devices might be used for particularly sensitive files, to help guard against unauthorized access. Backup copies of files may be more easily provided with separate, removable files. Sensitive data can be more easily removed – and thus protected – during program debugging periods.

Other factors

The "residue" problem can affect file design. The problem arises from two causes — the failure of a writing operation to completely erase what was previously recorded, as well as data that has not yet been overwritten. It can occur in main memory, on drums, disks, and on tape. (For that matter, any memory device — such as printer ribbons, printer platens, punched cards, paper tape, etc. — can cause a problem by retaining confidential information.) Ware (Reference 7) discusses tests that the Air Force made on the residue problem with magnetic tapes. Removable magnetic media, such as magnetic tapes, allow for degaussing to help remove residue. At the same time, removable media raise the spectre of theft — someone might "borrow" one or more magnetic tapes and read sensitive data in residue form.

Barron (in Reference 2) points out that safeguards must be designed into the system to protect against system breakdown at particularly critical times. If breakdown occurs while indexes are being updated, or security tables updated, records might be "lost" and hard to recover or invalid security checks might occur.

Also, A. E. Speckhard, in a letter to us, points out that long chains of pointers (connecting related data records) are vulnerable to hardware and software failures. Such failures can result in improper updating or inaccessible records — and correction might be more difficult under tight access control. The problem is not confined to long chains but applies to pointers of any kind.

Hardware/software techniques

All of the authorities that we talked to agree that the computer must have both read and write protect features for main memory, as a security measure. These features have other uses, of course, but are mandatory for an effective data security system.

Other desirable hardware features include the use of parity checks (present in most — but not all — computers) and the decoding of all undefined instruction codes. Undefined instructions might be tailored by a maintenance engineer, for instance, to provide a bypass around the security system.

The interrupt system used by the computer can cause security problems. Some interrupt systems can break into the middle of an operation — again possibly providing an opportunity for getting around safeguards.

Professor Glaser strongly recommends that the operating system run in the user's state (non-privileged state) as much as possible — say, 99.9% of the time. The privileged state, since it is unprotected, should be used as little as possible. Most current operating systems are not designed in this manner.

Dr. Ware, expanding on Glaser's idea, suggests going a step further. Why not, he says, separate the minimum necessary privileged mode from the operating system and put it in a second computer. This second computer would then handle *all* of the privileged functions such as security, audit trail, and input-output. Users would have no need to access this machine directly; in fact, hardware safeguards could be used to protect an intruder from getting into the privileged machine. Maintenance situations would have to be carefully controlled, of course.

Glaser suggests other features to promote security, in the design of the hardware/software complex. The system should be designed in compartments, rather than a monolithic design. "Concentric rings" of greater and greater privilege could be used. He suggests that some agency be established for *certifying* the data security system, so that users have some objective measure of their system's effectiveness. He raises the question, though: will the system have to be recertified each time the hardware or software is changed? Or whenever the operating system is recompiled? He points out that no user should be on line when the system is being compiled; otherwise, its integrity cannot be assured. Also, the system will have to be audited aperiodically, to make sure that it has not been changed.

Currently there is no recognized agency that performs such certification. Users who require secure systems have attacked this problem by having members of their staff try to penetrate the system safeguards. Or they have hired skilled "penetrators" to try to break the system.

A final note on operating systems. The two

systems that we encountered in our study that claim a high level of data security – the ADEPT-50 system and the National Security Agency system reported by Peters (Reference 2d) – both use specially designed operating systems. In fact, ADEPT-50 *is* an operating system. Neither attempted to patch up a manufacturer-developed operating system. (Which makes one wonder about the validity of the manufacturers' argument that operating systems should not be "unbundled.")

Communication protection

Earlier in this report, we discussed the threats to security that are posed by electromagnetic radiation and wiretaps. A user cannot assume that his communication lines are secure, if highly sensitive data is being transmitted or is in on-line files.

There are several types of countermeasures that can be taken to protect communications, with different costs and benefits. At the high cost end of the range is shielding – for the computer room, peripheral units, terminals, and even the communication lines themselves. Shielded lines can be tapped and thus need to be physically protected; thus they may be feasible only for local hard-wired lines.

If sensitive data is to be transmitted over common carrier circuits, some form of privacy transformation (encryption) may be needed. This subject is discussed by Petersen and Turn, Skatrud, Hoffman, and Baran (References 2b, 1e, 4, and 8); space does not allow an extended discussion here and the interested reader is referred to these references. Suffice it to say that available techniques include character substitution, character transposition, and the addition of one or more "keys."

Privacy transformation can also be used to protect file contents – file data would be stored in encryted form. Upon being read, it would have to be decoded, operated upon, recoded, and stored. While this idea has been proposed, Peters rejected it as "not worth the effort" during his remarks at the 1967 SJCC. Other people feel that it would provide a measure of security, particularly for data on demountable storage media.

Last month we discussed briefly the subject of data compaction, for use with data communications and file storage. Compaction is a form of encoding. It may be that compaction itself would provide some element of security – for instance, for less sensitive data.

Dedicated communication lines, either in the form of local hard-wired lines or lines leased from a common carrier can improve security. An intruder cannot gain access just by dialing in – but wiretaps are still very possible.

Finally, aperiodic checks can be made on communication lines, if wiretaps are considered possible. We understand that likely locations of the taps would be at junctions and line terminations between computer (or terminal) and the local telephone central office. Unusual line noise might also signal the presence of a tap.

Baran (Reference 8) discusses a proposed digital communication system of the future. He proposes, among other things, that *all* data be encrypted, with the more sensitive data receiving a higher level of protection. Further, he suggests that no pauses be allowed in data communications; the computer would transmit encrypted "noise" messages when it had no traffic. In this way, an intruder would be forced to pay a high price to get *any* useful information. Baran's ideas are most stimulating.

Ware, in his remarks at the 1967 SJCC (Reference 2a) stated that probably most communication safeguards will have to come from the people who *use* common carrier circuits and not from the common carriers themselves. One exception to this, pointed out to us by Glaser, is the Telephone Company's new Electronic Switching System (ESS). ESS is able to identify where a call is coming *from*, a big assist in tracking down intruders. Older switching mechanisms do not have this capability.

Reliability, auditability, integrity

Reliability includes guarding against system breakdown or error, either accidental or deliberate, and the ability to recover on a timely basis from either breakdown or error. Reliability is important for data security for two reasons. Failure may occur in the protection

funetion itself, opening the way for peneration. Also, protection may disappear during the recovery operations following a system failure; funny things happen during recovery.

Guarding against error is at least partially eovered by good internal eontrol proeedures, long advoeated by auditors. For instanee, the American Institute of CPAs has published a book on this subjeet. (Reference 9). Wasserman, in a short paper (Reference 10), gives a good summary. Records of performanee for men and machines are needed. Controls should be used for all input, output, and errors, with logs of all errors, operator interventions, machine halts, etc. A system of formal program change control should be used. Good security proeedures should be employed in the computer center — no admittance of unauthorized personnel, etc. Backup facilities should be available, and backup copies of files should be stored at a remote loeation.

Internal control proeedures such as these will eateh most of the accidental errors and at least some of the lower level deliberate intrusions.

One key system design feature mentioned by several people we talked to is the need for an extensive audit trail. For one thing, it is not possible to prevent *all* intrusions — so it would be desirable to have a record of what happened. If the system fails, then the safeguards may also fail — and it is not possible to prediet what will happen in all eases. When a system has just been installed, the error rate generally is high — and a record of all events will help detect system weaknesses. All of these reasons support the need for an extensive audit trail, in which all security events are recorded — what was changed, when it was changed, and who changed it.

Glaser suggests that the system be used to maintain itself. That is, ehanges to security tables, ehanges to security programs, etc. would not be entered directly. Rather they would be entered through the security system. Such changes would have the highest level of security privilege but would still be subjeet to reording in the audit trail. In fact, it might be desirable to require that all changes to the security system be entered jointly by two people.

Auditors and/or security personnel should eheck the integrity of the security system on an aperiodic basis, to see if the system has been changed. Kendall Wright of the Service Bureau Corporation favors the use of "hash" control totals for the security programs and tables. These totals probably would be maintained by the auditors or security personnel, outside of the system, and entered into the system during an integrity eheck. Such control totals are also useful during recovery from system breakdown, to insure that the security system has not been changed.

General security procedures

Both Baran (Referenee 8) and Peters (Referenee 2d) state a security principle that at first seems surprising. This principle is: if one cannot safely describe a proposed security system in the unclassified literature, then it is not sufficiently secure to be used with confidence. Peters used an analogy in his presentation; locks are described in detail in the patent literature, but this does not aid a burglar in breaking into a bank vault. A system gains its logical power by standing up under scrutiny. A system gains its proteetive power by providing such a large number of possible combinations that finding *the* combination is very difficult.

It is interesting to note that Clark Weissman's paper does just that — it deseribes the complete logical structure of a security system, as well as its implementation in ADEPT-50.

There are a number of tested security proeedures that can be used, if the value of the data warrants their use. One such proeedure is a security eheck on the background of people who might have aeeess to sensitive data. This would include remote terminal users, system programmers, operators, maintenance personnel, tape librarians, system managers, auditors, and security personnel.

The number of people who are authorized access to the data should be limited on a need-to-know basis. Some record of actual use of the data might be kept — for each authorized user, if he did, in fact, use the system. If a user has made no use of the system during a period of, say, six months, then perhaps he should be removed from the list of authorized users.

The position of security officer might be es-

tablished; such a person would oversee the whole security system. All non-trivial violations of safeguards would be reported to him – where an example of a trivial violation might be one bad attempt by a user to enter his password. Serious violations would be reported in real time, perhaps by way of a terminal in the security officer's office. If the offending terminal were located nearby, the security officer would go to it to find out what was going on. If the terminal were some distance away, he would call a supervisor at that location and have him check. File owners also should be told of these violations.

For highly sensitive data, it might be desirable to use a completely dedicated computer system, with no connection to remote terminals. Requests would come in to retrieval operators at the computer center. After such an operator was satisfied with the validity of the request, he would enter it on his console, read the response, and retransmit it to the remote user. This is the procedure followed by the New York State Identification and Intelligence System (NYSIIS) in Albany, New York.

Management should try to create a community of interest in the security system – for each user's self interest. Professor Glaser described a good example to us; it happened at Project MAC at M.I.T. One user suddenly noticed that he was getting output on his terminal that was not his. As he examined it, he realized that it was a list of the passwords used in the system. He didn't know what other users were receiving the same information – and he recognized that his own private files might be penetrated if a "wrong" person got his password. So he immediately entered a command that he knew would "crash" the system – as the best way, in this instance, to protect the data. He then called the center and told them what had happened. This is the sort of responsibility that should be encouraged in users, Glaser feels, not only for users at remote terminals but for all others who have some access to the data (system programmers, operators, maintenance personnel, etc.).

One of the key points in any security system is the assignment of responsibility. Someone must be made responsible for the security of each and every sensitive file. In a business environment, where many departments might be accessing the files – and particularly under the CDB concept – it will be difficult to make one person responsible. But unless one person *is* responsible, no one will be responsible – and the door is opened for penetration.

A related point involves the assignment of security classification to a new file. Rules must be established on who assigns the security level and what level should be assigned. The fact that this subject is difficult does not mean that it should be ignored. It is interesting to note that the ADEPT-50 system *automatically* assigns a security level to a new file created from one or more existing classified files. The user may override the system in assigning classification, but in the absence of any action by the user, a classification is assigned.

How to get a secure system

From our study of this subject of data security, we conclude that if your remote terminal system is typical of the vast majority of such systems today – that is, if:

- It uses conventional hardware, with read protect and/or write protect missing;
- It uses a manufacturer-supplied operating system – "so complex that one person cannot comprehend it";
- It uses a variety of regular terminals, either typewriter-like or CRT;
- The terminals are scattered over a wide geographical area;
- Communications are via common carrier circuits, either leased or dial network –

then

do not put any highly sensitive data on line to the computer – even if such files "are not supposed to be" accessible by the remote terminals – or accept the risk that the data may be divulged to unauthorized individuals.

What must you do if you want to put sensitive data in the on-line files? Dr. Ware and Professor Glaser gave the following suggestions:

- Limit the number and types of users that will use the system;
- Limit the number and the types of termi-

nals and the location of those terminals;

- Limit the sensitivity of the data in the files as much as possible, so as to limit the potential threats;
- Design and build "clean" software, particularly the operating system, and incorporate security features in the design;
- Be sure the hardware includes the essential features discussed earlier;
- Include techniques to safeguard the data, such as discussed in this report, that are appropriate to the value of the data.

If you are willing to so constrain your system, and to pay the development and operating costs, then it is likely that a system can be engineered with security sufficient to meet your needs.

Toward a corporate data base

Today's typical data processing department is already swamped with workload. Many installations are still converting second generation programs and files to third generation operation. They are converting new application systems as rapidly as they can. They are evaluating new types of hardware and software — optical scanning, computer-onto-microfilm, remote terminals, data management systems, and other software packages.

In such an environment, data processing management needs another big project such as the corporate data base like it needs a fire in the tape vault. In this series of four reports, we have tried to identify some of the problem areas:

- Getting agreement and support for company-wide standard data definitions;
- Learning to handle more complex file structures;
- Accommodating the growing bulk of stored data;
- Selecting and converting to a new data management system;
- Providing adequate security for sensitive data.

With challenging problems such as these, we suspect that most data processing managements will try to delay a CDB project as long as possible.

Nevertheless, pressures *do* exist for getting a CDB project under way. We discussed some of these pressures throughout this series of reports — pressures such as the need to shorten application system development cycles, reduce duplication of data, and provide better data compatibility. So, like it or not, a growing number of data processing managements may well find that they have to start a CDB project.

Some companies will choose to do the job thoroughly — involving a large effort and a long range program. Unless care is taken to do otherwise, it may take several years before benefits begin to show up with such a program. But once they do begin to appear, they should start to flow rapidly.

We suspect that most companies will want to make a more modest effort and achieve some benefits sooner. We think that useful results *can* be obtained within, say, one year, if the project is planned that way. The characteristics of such a project would be:

Create an inventory of current data definitions in the mechanized system. A cross-referenced inventory of current fields, records, files, and programs is a first step toward standardized data definitions.

Start using a commercial data management system. Despite their limitations, today's data management systems can be very useful. For one thing, such a data management system can be a big help in creating and maintaining the inventory of data definitions just mentioned. Also, such a system can be used to convert at least the simpler application systems to the computer — and it can aid in converting to new equipment. If you choose to build your own data management system, to get more of the features you desire, you probably will not be able to gain many benefits within one year.

Go easy on file expansion. Today's data management systems are fairly limited in the types of storage structures they can handle — with a very few notable exceptions. As data files are expanded by adding new types of data, more complex structures result. Trying to handle these more complex structures with most of to-

day's data management systems can put a roadblock in your path.

Be very cautious about putting sensitive data on-line. As we have seen in this issue, data security is a complex subject. If your project gets deeply involved with it, the project schedule will be affected.

As we said at the outset of this series, the corporate data base seems to be an emerging trend in the field. We hope this series of reports has put the CDB benefits and problems into perspective.

REFERENCES

1. *Proceedings of the 1969 Fall Joint Computer Conference,* published by AFIPS Press (210 Summit Avenue, Montvale, N. J. 07645), price $25 (microfiche $10):
   a) Weissman, Clark, "Security controls in the ADEPT-50 time sharing system"
   b) Linde, R. R., C. Weissman, C. Fox, "The ADEPT-50 time sharing system"
   c) Skatrud, R. O., "The application of cryptographic techniques to data processing"
2. *Proceedings of the 1967 Spring Joint Computer Conference,* AFIPS Press (address above), price $20.20; microfiche version may be available:
   a) Ware, W. H., "Security and privacy in computer systems"
   b) Petersen, H. E. and R. Turn, "System implications of information privacy"
   c) Barron, D. W., "File handling at Cambridge University"
   d) Peters, B., "Security consideration in a multiprogrammed computer system"
3. California Senate Judiciary Committee, "The interception of messages by the use of electronic and other devices . . ." 1957 Regular Session, California Legislature.
4. Hoffman, L. J., "Computers and Privacy: A Survey," *Computing Surveys* (ACM, 1133 Avenue of Americas, New York, N.Y. 10036), June 1969, p. 85-103; price $5.
5. *The Considerations of Data Security in a Computer Environment,* IBM Corporation, Form No. 520-2169; contact local IBM office.
6. Hsiao, D. K., *A file system for a problem solving facility,* doctoral thesis, University of Pennsylvania, 1968; order from University Microfilms, Inc., Ann Arbor, Mich. 48106; 156 p., price $9.
7. Ware, W. H., in "Security in the computer environment," System Development Corporation, August 1966; order from Clearinghouse for Federal Scientific and Technical Information (Springfield, Va. 22151), No. AD 640 648; 34 p.; price $3 hardcopy, 65¢ microfiche.
8. Baran, Paul, *On Distributed Communications: IX Security, Secrecy, and Tamper-free Considerations,* The Rand Corporation; August 1964; order from the Clearinghouse (address and prices above), AD 444 837.
9. Davis, G. B., *Auditing and EDP,* American Institute of Certified Public Accountants (666 Fifth Avenue, New York, N.Y. 10019), price $12.
10. Wasserman, J. J., "Plugging the leaks in computer security," *Harvard Business Review* (Soldiers Field, Boston, Mass. 02163), September-October 1969, p. 119-129, price $1.

Other reports on data security

11. Bingham, H. W., "Security techniques for EDP of multilevel classified information," order from Clearinghouse (address and prices above), AD 476 557.
12. Van Tassel, D., "Advanced cryptographic techniques for computers," *Communications of the ACM* (ACM, address above), December 1969, p. 664-665. Also see his paper in the *Proceedings of the 1969* SJCC (AFIPS Press, address above), p. 367-372; his article in the *Journal of Systems Management* (24587 Bagley Road, Cleveland, Ohio 44138), February 1969; and his article in *Computers and Automation* (815 Washington St., Newtonville, Mass. 02160), July 1969.
13. Guise, R. F., "File Security," *Data Systems News* (200 Madison Ave., New York 10016), November 1969, p. 60.

Data entry and data output have long been bottleneck operations for EDP. There has been quite a bit of progress in the data entry sector — via optical scanning, key-to-tape devices, etc. Now computer output to microfilm (COM) is finally catching on as a practical output method, after more than a decade of incubation. Next month we will discuss some of the progress, benefits, and costs of COM — and how it is even competing with some on-line inquiry systems.